PATENT
Atty. Dkt. No. 1475.013272 (AUTO/0034)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: John Wainright	§	Confirmation No.: 1474
	§	
	§	Group Art Unit: 2628
Serial No.: 09/426,143	§	
	§	Examiner: Chante E. Harrison
Filed: October 22, 1999	§	
	§	
For: SPECIFYING OPERATIONS TO BE APPLIED TO THE ATTRIBUTES OF A SET OF OBJECTS	§	

MAIL STOP APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Chante E. Harrison, or electronically transmitted via EFS-Web, on the date shown below:

October 15, 2007 /Jon K. Stewart/
Date Jon K. Stewart

RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF DATED SEPTEMBER 13, 2007

Dear Sir:

Appellants submit this Substitute Appeal Brief to the Board of Patent Appeals and Interferences in response to the Notification of Non-Compliant Appeal Brief dated September 13, 2007 in the Appeal of the above-identified application.

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to Autodesk, Inc., 111 McInnis Parkway, San Rafael, California 94903.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-3, 5, 7-14, 16 and 18-22 are pending in the application. Claims 1-20 were originally presented in the application. Claims 4, 6, 15 and 17 have been canceled without prejudice. Claims 1-3, 5, 7-14, 16 and 18-22 stand finally rejected as discussed below. The final rejections of claims 1-3, 5, 7-14, 16 and 18-22 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments include a method (see, e.g., claim 1) for executing an operation on a set of graphical components. See, e.g., *Application*, 1:5-7, 6:1-11, 12:3-17, and, Abstract. The graphical components generally represent the shapes and surface that make up a modeled entity. See, e.g. *Application* 1:15-20, 11:20-25.

As recited by claim 1, the steps of the method include detecting that a statement contains certain information. See, e.g., *Application*, 8:10-19. Specifically, the method includes detecting that a statement contains an operation identifier that specifies said operation to be performed on the set of graphical components. See, e.g., *Application*, 6:1-9, 8:10-11, 10:2-17, 10:18-24 – 11:1-2, 13:1-7, 14:6-9, 16:1-3. The detected statement also contains pattern matching criteria used to identify the set of graphical components. See, e.g., *Application,* 8:12-13, 12:20-25, 13:1-25, 14:1-5. Lastly, the detected statement contains an attribute identifier that identifies an attribute. See, e.g., *Application*, 8:14-15, 9:3-8.

As recited by claim 1, the steps of this method also include executing the detected statement by identifying said set of graphical components associated with identifiers that satisfy said pattern matching criteria. See, e.g., *Application*, 8:13-14, 13:14-19, 15:21-23, 17:7-11. The steps of this method also include performing said operation on said attribute of each graphical component in said set of graphical components that satisfy said pattern matching criteria, altering state information corresponding to each graphical component in said set of graphical components to generate a frame within an animation. See, e.g., *Application*, 8:14-20, 13:14-19, 15:21-23, 17:7-11.

Claimed embodiments also include a method (see, e.g., claim 8) for executing an operation on a set of graphical components. As recited by claim 8, the steps of this method include detecting that a statement contains certain information. See, e.g., *Application*, 8:10-19. Specifically, this method includes a step of detecting that a statement contains an operation identifier that specifies the operation to be executed on a set of graphical components. See, e.g., *Application*, 8:10-11, 10:2-17, 10:18-24 – 11:1-2, 13:1-7, 14:6-9, 16:1-3. The statement detected also contains an identifier that is associated with a collection of graphical components See, e.g., *Application* 6, 9-11, 16:1-19. 14:5-12 – 15:1-6. The statement detected also contains an attribute

identifier that identifies an attribute of a member graphical component of said collection of graphical components; See, e.g., *Application*, 8:14-15, 9:3-8.

As recited by claim 8, the steps of this method also include executing said statement by identifying member graphical components of said collection of graphical components. See, e.g., *Application*, 8:13-14, 13:14-19, 15:21-23, 17:7-11. As recited by claim 8, the step of executing said statement also includes performing said operation on said attribute of each graphical component of said identified member graphical components, altering state information corresponding to each graphical component of said identified member graphical components to generate a frame within an animation. See, e.g., *Application*, 8:14-20, 13:14-19, 15:21-23, 17:7-11.

Claimed embodiments also include a computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on a set of graphical components, (see e.g., claim 12). See, e.g., *Application*, 1:5-7, 6:1-11, 12:3-17, and, Abstract. The graphical components generally represent the shapes and surface that make up a modeled entity. See, e.g. *Application* 1:15-20, 11:20-25. The instructions include instructions which, when executed by one or more processors, cause the one or more processors to perform a set of steps. As recited by claim 12, the steps include detecting that a statement contains certain information. See, e.g., *Application*, 8:10-19. Specifically, the steps include detecting that a statement contains an operation identifier that specifies said operation to be performed on the set of graphical components. See, e.g., *Application*, 6:1-9, 8:10-11, 10:2-17, 10:18-24 – 11:1-2, 13:1-7, 14:6-9, 16:1-3. The detected statement also contains pattern matching criteria used to identify the set of graphical components. See, e.g., *Application,* 8:12-13, 12:20-25, 13:1-25, 14:1-5. Lastly, the detected statement contains an attribute identifier that identifies an attribute. See, e.g., *Application*, 8:14-15, 9:3-8.

As recited by claim 12, the steps also include executing the detected statement by identifying said set of graphical components associated with identifiers that satisfy said pattern matching criteria. See, e.g., *Application*, 8:13-14, 13:14-19, 15:21-23, 17:7-11. The steps also include performing said operation on said attribute of each graphical component in said set of graphical components that satisfy said pattern matching criteria, altering state information corresponding to each graphical component in said set of graphical components to generate a frame within an animation. See, e.g., *Application*, 8:14-20, 13:14-19, 15:21-23, 17:7-11.

Claimed embodiments also include a computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on collections of graphical components, (see e.g., claim 18). See, e.g., *Application*, 1:5-7, 6:1-11, 12:3-17, and, Abstract. The graphical components generally represent the shapes and surface that make up a modeled entity. See, e.g. *Application* 1:15-20, 11:20-25. As recited by claim 18, the instructions include instructions which, when executed by one or more processors, cause the one or more processors to perform a set of steps. As recited by claim 18, the steps include detecting that a statement contains certain information. See, e.g., *Application*, 8:10-19. Specifically, the steps include a step of detecting that a statement contains an operation identifier that specifies the operation to be executed on a set of graphical components. See, e.g., *Application*, 8:10-11, 10:2-17, 10:18-24 – 11:1-2, 13:1-7, 14:6-9, 16:1-3. The statement detected also contains an identifier that is associated with a collection of graphical components See, e.g., *Application* 6, 9-11, 16:1-19. 14:5-12 – 15:1-6. The statement detected also contains an attribute identifier that identifies an attribute of a member graphical component of said collection of graphical components; See, e.g., *Application*, 8:14-15, 9:3-8.

As recited by claim 18, the steps also include executing said statement by identifying member graphical components of said collection of graphical components. See, e.g., *Application*, 8:13-14, 13:14-19, 15:21-23, 17:7-11. As recited by claim 18, the step of executing said statement also includes performing said operation on said attribute of each graphical component of said identified member graphical components, altering state information corresponding to each graphical component of said identified member graphical components to generate a frame within an animation. See, e.g., *Application*, 8:14-20, 13:14-19, 15:21-23, 17:7-11.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1-3, 5, 7-14, 16 and 18-22 are rejected under 35 § U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20020008703 to John Merrill, et al. (hereinafter "*Merrill*").

ARGUMENTS

A. Claims 1-3, 5, 7-14, 16 and 18-22 are not obvious in view of *Merrill*

Independent claims 1 and 8 (along with corresponding computer readable medium claims 12 and 18) stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Merrill.* Respectfully, Applicant believes that *Merrill* fails to teach to teach or suggest the limitations of the present claims.

Merrill is generally directed to the synchronization of interactive animations. That is, animations responsive to user input. See *Merrill*, Abstract. For example, an animation that includes a speaking character must be synchronized so that movements of a mouth correspond to the audio of the animation. The user input may be to begin, stop, or pause an animation sequence. Further, such an animation may need to be synchronized with other, autonomous characters displayed as part of an animation sequence to give the appearance of characters speaking to one anther. See *Merrill*, ¶ 3, 4, 10.

To create the illusion of speech in an animation, Merrill discloses using an animation server to draw a bitmap representing the mouth at the (x, y) coordinates where the character's mouth is located.

> In this implementation, the servers maintains bitmaps representing the character's mouth in a variety of different positions in a mouth animation file. There are a number of different bitmaps, each representing the position of the mouth for a corresponding phoneme. To enhance realism, the server can maintain different sets of mouth data files and select the appropriate one based on the position of the character. Each set of mouth data files can map a set of phonemes to bitmaps representing the mouth position for a phoneme.
>
> To lip-synch the mouth animation with the speech output, the server instructs the speech synthesis engine to notify it before it generates speech output for a phoneme. Just before the speech synthesis engine is about to output a phoneme, it passes a message to the mouth animation module identifying the phoneme. The mouth animation module loads the animation and draws at the (x,y) location on top of the current frame of animation.

Merrill, ¶ 87-88. *Merrill* also discloses the use of "special" markup tags embedded in an HTML web page tag to synchronize the operations of a text-to-speech engine with the display of an animation sequence. See *Merrill*, ¶ 167-171.

Merrill also discloses the use of an animation server to present a synchronized animation sequence to a client. See *Merrill*, ¶ 146-171. *Merrill* goes on to describe "specific examples of the methods properties and events of the objects implemented in the animation server." *Merrill*, ¶ 154 In addition to the programmatic aspects of the animation server, *Merrill* discloses that a client may interact with the animation server using known techniques for client server communications including COM (Common Object Model) and OLE (Object linking and embedding). Using these techniques, a client may communicate with the animation server to provide the user to request, receive, display and control one or more animation sequences.

Despite the focus in *Merrill* on the synchronization of animation sequences, the Examiner maintains that *Merrill* renders the present claims unpatentable. Applicant respectfully disagrees.

In rejecting independent claims 1, 8, 12, and 18, the Examiner cites a number of disconnected passages from *Merrill* that fail to disclose the limitations recited the present claims. For example, the Examiner asserts that "Merrill discloses detecting that a statement contains an operation identifier, pattern-matching criteria, and attribute identifier … (pp. 13, Para 168-169; pp. 19, Para 324-327)." *Final Office Action,* p. 2 and *Advisory Action*, continuation sheet. However, the first passage cited by the examiner is in fact directed to an example of a "special type of tag called a bookmark tag in Speak method statement [sic] to sync its operations with the output text." *Merrill*, ¶ 168. That is, to an HTML markup tag that may be embedded as part of a web-page. Nowhere in passage is any pattern matching criteria used to identify a set of graphical components.

The "special" HTML tags are used to synchronize text-to-speech aspects of the system described above. For example, immediately prior to the passage cited by the Examiner, *Merrill*, provides a general description of these "special" HTML tags:

> The server also embeds what are referred to as tags in every piece of text that is passed to the speech synthesis engine. These tags are inserted before every word in the text and tell the speech synthesis engine that the server wants to be notified whenever one of the tags is encountered. The server can then use this data to display the word that is currently being spoken in a visual user interface. This technique can then be used effectively to close caption the text as it is being spoken. In this implementation the server displays this text in a graphic representing a balloon.

Merrill, ¶ 167. Candidly, the passage cited by the examiner lacks any teaching of pattern matching criteria (as recited by claims 1 and 12) or of an identifier that is associated with a collection of graphical components (as recited by claims 8 and 18). Rather, the passage is directed to special tags embedded in a block of text fed to a text-to-speech generator. The problem addressed in these passages of *Merrill* is the synchronization of the movements of a character's mouth appearing in an animation sequence.

Further, the second passage cited by the Examiner, merely provides exemplary syntax of Microsoft's Visual Basic product. Nothing in this second passage teaches a statement that contains pattern matching criteria; rather, the passage provides some simple examples of Visual Basic. Respectfully, Applicant submits that *Merrill* a general description Visual Basic syntax fails to teach the claimed limitation of detecting a statement that includes (i) an operation identifier that specifies said operation, (ii) pattern matching criteria, and (iii) an attribute identifier that identifies an attribute, as recited in Claims 1 and 12. There is simply no pattern matching criteria present in these general examples of Visual Basic syntax.

Moreover, the Examiner asserts that *Merrill* discloses "executing the statement by identifying said set of graphical components associated with identifiers that satisfy pattern matching criteria. (pp. 20, Para 340)." *Final Office Action*, p. 2 and *Advisory Action*, continuation sheet. However, the text cited by the Examiner regarding this limitation provides:

> As the browser renders the Web page, it also encounters the script. For Visual Basic Script, the browser loads a Visual Basic Script runtime interpreter locally to translate the Visual Basic script on-the-fly and run the code. If the browser supports other scripting languages, it loads the appropriate interpreter based on the script language identified in the document. When the browser encounters script code, it loads an appropriate interpreter for the script language, and this interpreter then translates the code. The script code executes via calls from the interpreter in response to references to the character control interface, which in this specific implementation is the OLE control interface described in detail above. In the specific case of Visual Basic Script, for example, the browser loads an interpreter in the process space of the browser. To execute the script code, the browser uses the interpreter to translate the code and then accesses the OLE control interface in response to references to the control interface in the script code. As noted above, the browser loads the OLE control representing the character into the process space of the browser when it encounters an object identifier called the object tag. Thus, in this particular implementation, both the control and the interpreter are loaded in the process space of the browser. When the script code

> references the character control, the browser accesses the animation server, which runs in a separate process, through the control interface. The control acts a gateway, routing requests for access to the animation server's methods and properties for a particular character to the animation server.

Merrill, ¶ 340.

Nothing in this passage teaches or suggests executing the detected statement by *identifying* a set of graphical components that satisfy pattern matching criteria, as recited by claims 1 and 12, or by *identifying* member graphical components of said collection of graphical components, as recited by claims 8 and 18. Rather, the passage is directed to a method for processing a script included in the source of an HTML based webpage to invoke features of the animation server of *Merrill*. When a web browser encounters a tag indicating the beginning of a script configured to interact with the animation server disclosed in *Merrill*, the contents of the script are provided to an appropriate process that can perform the sequence of instructions defined for the script. Typically, the script is used to present a user with a synchronized animation sequence where the audio is synchronized with the movements of an animated mouth. As described by this passage, the script interpreter is managed using Object Linking and Embedding (OLE). As is known by those of ordinary skill in the art, OLE is a distributed object system and protocol developed by Microsoft Corp. OLE allows one application (in this case a web browser) to "farm out" part of a document (in this case the animation script) to another editor (in this case the animation server disclosed by *Merrill*) and then re-import it. The recited elements of identifying a set of graphical components, either using pattern matching criteria, or an identifier, is simply not addressed.

Lastly, as *Merrill* fails to teach or suggest the limitations as described herein, *Merrill* further fails to teach or suggest the limitation of "performing said operation on said attribute of each graphical component in sad set of graphical components that satisfy said pattern matching criteria."

Accordingly, for all the foregoing reasons, Applicant submits that *Merrill* fails to teach or suggest the limitations recited in independent claims 1, 8, 12, and 18. Therefore, Applicant believes that claims 1, 8, 12, and 18, along with the claims dependent therefrom, are in condition for allowance, and respectfully request that the rejections be withdrawn.

CONCLUSION

The Examiner errs in finding that claims 1-3, 5, 7-14, 16 and 18-22 are unpatentable over *Merrill* under 35 U.S.C. § 103(a). Withdrawal of the rejection and allowance of all claims is respectfully requested.

Respectfully submitted,

/Jon K. Stewart/

Jon K. Stewart
Registration No. 54,945
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellant

CLAIMS APPENDIX

1. (Previously Presented) A method of executing an operation on a set of graphical components, the method comprising the computer-implemented steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

pattern matching criteria, and

an attribute identifier that identifies an attribute; and

executing said statement by

identifying said set of graphical components associated with identifiers that satisfy said pattern matching criteria, and

performing said operation on said attribute of each graphical component in said set of graphical components that satisfy said pattern matching criteria, altering state information corresponding to each graphical component in said set of graphical components to generate a frame within an animation.

2. (Original) The method of Claim 1, wherein said statement includes a first string of characters that contains at least one wild card character and that specifies said pattern matching criteria.

3. (Original) The method of Claim 2, wherein said first string is part of a second string of characters, wherein said second string of characters includes said attribute identifier and is in a format that conforms to object-dot notation.

4. (Canceled)

5. (Original) The method of Claim 1, wherein said statement is written in a scripting language and the step of detecting is performed by a script processor.

6. (Canceled)

7. (Original) The method of Claim 1, wherein step of detecting that a statement contains pattern matching criteria includes detecting that the statement contains pattern matching criteria for a hierarchical identifier.

8. (Previously Presented) A method of executing an operation on collections of graphical components, the method comprising the computer-implemented steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

an identifier that is associated with a collection of graphical components, and

an attribute identifier that identifies an attribute of a member graphical component of said collection of graphical components; and

executing said statement by

identifying member graphical components of said collection of graphical components, and

performing said operation on said attribute of each graphical component of said identified member graphical components, altering state information corresponding to each graphical component of said identified member graphical components to generate a frame within an animation.

9. (Previously Presented) The method of Claim 8, wherein said collection of graphical components is an array.

10. (Previously Presented) The method of Claim 8, wherein said collection of graphical components includes all instances of a native type of graphical components managed by a CAD system.

11. (Original) The method of Claim 10, wherein said native type is a map type of graphical components, wherein a map type defines a surface.

A method of executing an operation on a set of graphical components

12. (Previously Presented) A computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on a set of graphical components, the one or more sequences of one or more instructions including instructions which , when executed by one or more processors, cause the one or more processors to perform the steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

pattern matching criteria, and

an attribute identifier that identifies an attribute; and

executing said statement by

identifying all graphical components associated with identifiers that satisfy said pattern matching criteria, and

performing said operation on said attribute of each of said graphical components that satisfy said pattern matching criteria, altering state information corresponding to each graphical component in said set of graphical components to generate a frame within an animation.

13. (Original) The computer-readable medium of Claim 12, wherein said statement includes a first string of characters that contains at least one wild card character and that specifies said pattern matching criteria.

14. (Original) The computer-readable medium of Claim 13, wherein said first string is part of a second string of characters, wherein said second string of characters includes said attribute identifier and is in a format that conforms to object-dot notation.

15. (Canceled)

16. (Original) The computer-readable medium of Claim 12, said statement is written in a scripting language and the step of detecting is performed by a script processor.

17. (Canceled)

18. (Previously Presented) A computer-readable medium carrying one or more sequences of one or more instructions for executing an operation on collections of graphical components, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

detecting that a statement contains

an operation identifier that specifies said operation,

an identifier that is associated with a collection of graphical components, and

an attribute identifier that identifies an attribute of a member object of said collection of graphical components; and

executing said statement by

identifying member graphical components of said collection of graphical components, and

performing said operation on said attribute of each graphical component of said identified member graphical components, altering state information corresponding to each graphical component of said identified member graphical components to generate a frame within an animation.

19. (Previously Presented) The computer-readable medium of Claim 18, wherein said collection of graphical components is an array.

20. (Previously Presented) The computer-readable medium of Claim 18, wherein said collection of graphical components includes all instances of a native type of graphical components managed by a CAD system.

21. (Previously Presented) The method of claim 1, further comprising the step of changing the value of another attribute, the other attributes not associated with the identifiers that satisfy said pattern matching criteria.

22. (Previously Presented) The method of claim 8, further comprising the step of changing the value of another attribute, the other attribute not associated with the attribute identifier.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.